NISTIR 5512

A New Method to Calculate Looming for Autonomous Obstacle Avoidance

Kunal Joarder
Robotics Center and
Department of Electronics Engineering
Florida Atlantic University
Boca Raton, FL 33431

Daniel Raviv
Robotics Center and
Department of Electronics Engineering
Florida Atlantic University
Boca Raton, FL 33431
and Intelligent Systems Division

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Bldg. 220 Rm. B124
Gaithersburg, MD 20899

NIST

NISTIR 5512

A New Method to Calculate Looming for Autonomous Obstacle Avoidance

Kunal Joarder
Robotics Center and
Department of Electronics Engineering
Florida Atlantic University
Boca Raton, FL 33431

Daniel Raviv
Robotics Center and
Department of Electronics Engineering
Florida Atlantic University
Boca Raton, FL 33431
and Intelligent Systems Division

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Bldg. 220 Rm. B124
Gaithersburg, MD 20899

November 1994

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arati Prabhakar, Director

A New Method to Calculate Looming

for

Autonomous Obstacle Avoidance

Kunal Joarder[1] and Daniel Raviv[1,2]

[1]Robotics Center and Department of Electrical Engineering
Florida Atlantic University
Boca Raton, Florida 33431; and
[2]Intelligent Systems Division
National Institute of Standards and Technology (NIST)
Bldg. 220, Room B124, Gaithersburg, MD 20899.

Abstract

The idea of visual looming, i.e., the expansion of an object's projected size on the retina due to its decreasing distance from the observer, can be used as a powerful cue for autonomous obstacle avoidance. In this paper we describe a method of measuring looming quantitatively by fixating a camera at a point on the surface of an object and studying the change in the visible texture near this point. A reduced distance of the camera from the surface usually results in a decline in the density of the texture primitives in the image. We show analytically how looming can be calculated from the relative change in this density and from the local orientation of the surface. The local orientation is simply obtained from a set of one-dimensional directional densities of the texture primitives. The method is simple, uses a single camera and actively controlled visual fixation. It is applicable for both stationary and moving obstacles and does not require calculation of range. Results of looming are presented for various textural surfaces, in different orientations. This visual cue is used in a closed loop to implement obstacle avoidance using a six-degree-of-freedom simulator.

Key words: Obstacle Avoidance, Active Vision, Looming, Fixation, Texture, Visual Motion, Time-to-contact.

This research was supported in part by a grant to Florida Atlantic University from the National Science Foundation, Division of Information, Robotics and Intelligent Systems (Grant # IRI-9115939).

1. Introduction

Visual looming effect pertains to an increasing size of the projection of a 3-D object on a viewer's retina as the relative distance between them decreases. Psychologists have studied this phenomenon by observing vision and action in unison and have reported subjects' tendency to react defensively or using this information in an anticipatory control of the body. Beek [1] and Bower *et al.* [2] studied the effect of visual looming on infants and presented results in which infants attempted to avoid an enlarging image on a screen or for smaller objects it operated as an anticipatory system, preparing the hand for grasping. Schiff *et al.* [3] concluded that the looming effect was a strong exciter of avoidance for rhesus monkeys. The visual stimulus indicated an impending collision. Lee *et al.* [4] and Savelsbergh *et al.* [5] studied how the human motor systems are regulated in hitting an approaching ball or grasping an object based on this visual information. This basic perception-action coupling is based on a single visual stimulus, the so called looming. If looming can be defined quantitatively, then one can adopt the same approach for autonomous mobile robots for visual navigation tasks such as obstacle avoidance.

In this paper a practical way to measure visual looming quantitatively from a sequence of images obtained from a camera is described. The camera fixates at a point on the visible surface of an object as the relative distance between the camera and the object changes. The important parameter that is used in this approach is the variation of texture density in the image due to relative motion. As mentioned in [6], a uniformly textured surface undergoes two types of projective changes during the imaging process:

1. A decrease in the distance from the surface to the camera causes a decrease in the texture density in the image.
2. Due to the angle between the surface and the image plane, foreshortening results in an anisotropic texture density in the image.

Based on this we developed a closed form analytical expression that shows the following:

1. How the texture density in the image varies during fixated motion as the relative distance and the slant of the surface (angle between the surface normal and the optical axis of the camera) change.
2. How looming can be calculated from the relative change in the texture density in the image and the slant of the surface, both obtained from the image.

Looming is a concept related to, but different from, time-to-contact. As defined in [7] time-to-contact deals with the relative change in *depth* (i.e., the projection of the range along the motion vector of the observer) where looming deals with relative change in *range*. Points which are far away from the motion vector and not relevant to a decision for obstacle avoidance may still produce high values of time-to-contact. These points will produce small looming values. Another related concept is image divergence. Nelson *et al.* [8,9] used image flow information to determine the motion parameters of a moving observer and subsequently calculated the flow field divergence to implement autonomous obstacle avoidance. Cipolla and Blake [10] used image divergence and deformation to calculate surface orientation and time-to-contact. Image divergence, which either needs calculation of image flow or the moments of an image area is a way for calculation of looming when the surface is not tilted with respect to the motion vector. Looming can be extracted by studying several simple properties of an image, e.g., blur, brightness or texture density, as well as divergence.

Conceptually the approach presented here is also related to [11,12]. Tistarelli *et al.* [11] uses visual fixation and specially designed log-polar retina that considerably reduce the information needed to avoid collision by means of a time-to-contact map. Murphy [12] measured the change in visible texture for positioning mobile robots.

The proposed approach of calculating looming and using it for obstacle avoidance is simple since it uses only a single camera and does not require calculation of optical flow. It is not necessary to know the range of the object or the translational and rotational parameters of the camera. The approach is texture independent (as long as it is visible and finite). It is applicable for both stationary and moving objects and the camera can experience a general fixated motion. By maintaining its point of fixation on the visible surface of the object the camera studies the change in texture density in a small window in the image around this point and calculates looming from the relative change in texture density and the slant of the surface, both obtained from the image. This approach follows the active vision [13,14] paradigm and shows that if instead of reconstructing the complete three-dimensional scene in front of the camera, if one is able to supply only the useful information to the system, the computational expense is reduced significantly. In an active perception-action loop the camera can manipulate its points of attention to only the pertinent parts of the objects on its path and thus can reduce the vast visual field into small working domain, calculate looming from the change in image texture and use this variate as a threat to navigate between obstacles.

First, a brief review of looming is provided. It is shown that looming along with visual fixation simplifies the task of obstacle avoidance considerably. Next, it is shown how looming can be calculated by fixating a camera on the texture of a surface. Finally we present results that illustrate this new method and its use in obstacle avoidance.

2. Looming - A Review

2.1 Definition of Looming

A quantitative definition of looming can be found in [15]. The looming value L of an infinitesimally small 3D object is defined as the negative value of the time derivative of the relative distance (range) R between the observer and the center of the object divided by the relative distance :

$$L = -\frac{\frac{dR}{dt}}{R} \qquad \text{... (1)}$$

The negative sign is used to signify image expansion with positive looming. The unit by which looming is measured is [time $^{-1}$]. It has been shown in [15] that looming is independent of rotation of the camera.

2.2 Looming and fixation

In an Active Vision paradigm a moving camera actively manipulates its point of attention. For example, in calculating looming by directing its fixation point only to some small regions that are part of a bigger object, an autonomous system can reduce a vast three-dimensional space into small working domains and the computational expense is significantly reduced. Also the camera can conveniently have a high resolution but narrow field of view.

Suppose we have a camera moving towards a big stationary object (Figure 1). With a decreasing range the object should have an increasing looming value. The camera, instead of attempting to span the entire object using a large field of view, can locate some feature points on the object and fixate on them. Once it has fixated on a feature point the system can concentrate in a small window around this point and calculate the looming in that small local working area only. The locations of these feature points are relevant to the task being performed. For example, they may lie on a landmark for a navigation system or on an object on the path of the camera for an obstacle avoidance system.

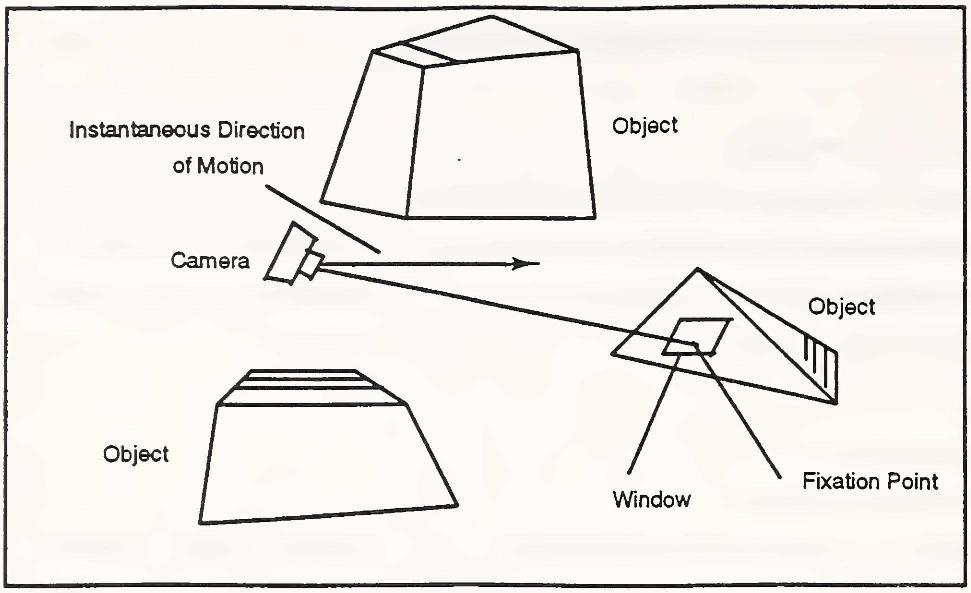


Fig. 1 Fixation point on an object

3. Measurement of Looming

3.1 Surface texture - the motivation

Here we present an approach in which the looming is calculated from the density and its temporal change of the image texture primitives which results from a relative motion between the camera and the textured surface. In the past it has been shown [15] that the relative rate of expansion of an object in the image is proportional to the looming value. Using this simple method an obstacle avoidance algorithm was implemented [16]. In this paper we suggest measurement of looming based on an extension of the texture related work in [17].

During the imaging process a uniformly textured surface experiences two types of projective changes [6]:

1. As the distance between the surface and the camera decreases a reduction of image texture density occurs on a fixed area of the image (image texture density is defined as the number of texture primitives or texels per unit area)
2. Foreshortening causes an anisotropic compression of the texture.

These two cues provide significant information about the relative distance of the surface and its orientation with respect to the camera. In the past most of the work has been conducted in the area of determining the shape of the surface from these cues. See for example [18-23]. In this paper the main emphasis is not on calculating the surface orientation or the absolute range (or depth) of the object, but to determine the *relative* *change* in the range, solely based on the image sequence without physically measuring the range or the orientation of the surface. Visual looming is subsequently calculated from the relative change in the range. For a vision-based task of obstacle avoidance looming as a cue is simple and sufficient.

In the present approach the texture primitives are described in terms of the brightness gradients in a region. The measurement involves the number of brightness gradients, which essentially signify the edgeness, in a certain region or along a certain line and these are considered the regional or the one-dimensional directional texture density, respectively. The camera is always fixated on a feature point on the textural surface and a small neighborhood around this point on the perspective projection image is selected to calculate the regional and the directional texture densities. The assumption here is that the small area around the point of fixation is locally planar and the density of texture primitives - i.e., the number of texture primitives per unit area - on the 3-D surface is constant in that region.

3.2 Texture density and surface normal

Here we provide a brief perspective of the spatial analysis portion of the derivation of texture as used by Haralick *et al.* in [17]. The analysis is done in a camera centered coordinate system. The camera pinhole point lies at the origin and forms a solid angle Ω at the neighborhood on the textural surface. Let us assume that the image coordinate system has its origin at the center of the image. Let the position of the center of the neighborhood on the image be given by (x,y). The unknown plane where the textural surface is observed can be expressed by :

$$AX + BY + CZ + D = 0 \qquad \text{... (2)}$$

where

$$A^2 + B^2 + C^2 = 1$$

and *A, B, C* are the direction cosines of the plane (Figure 2).

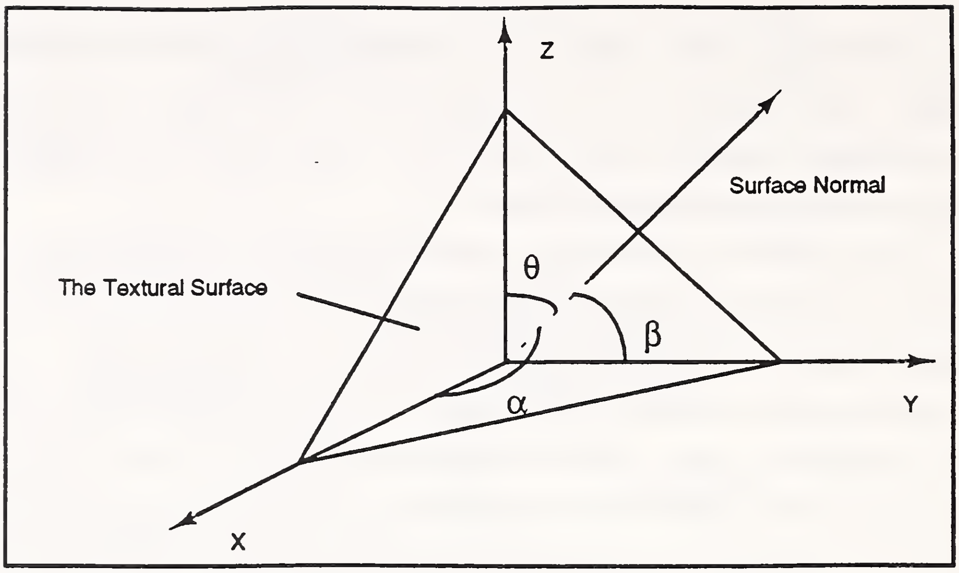


Fig. 2 The textural surface

$A = \cos \alpha$, $B = \cos \beta$, $C = \cos \theta$.

Following the convention of perspective projection in [17] the image is kept in a positive orientation. The focal length of the camera is f. The image plane is located in front of the pinhole point of the camera as in Figure 3.

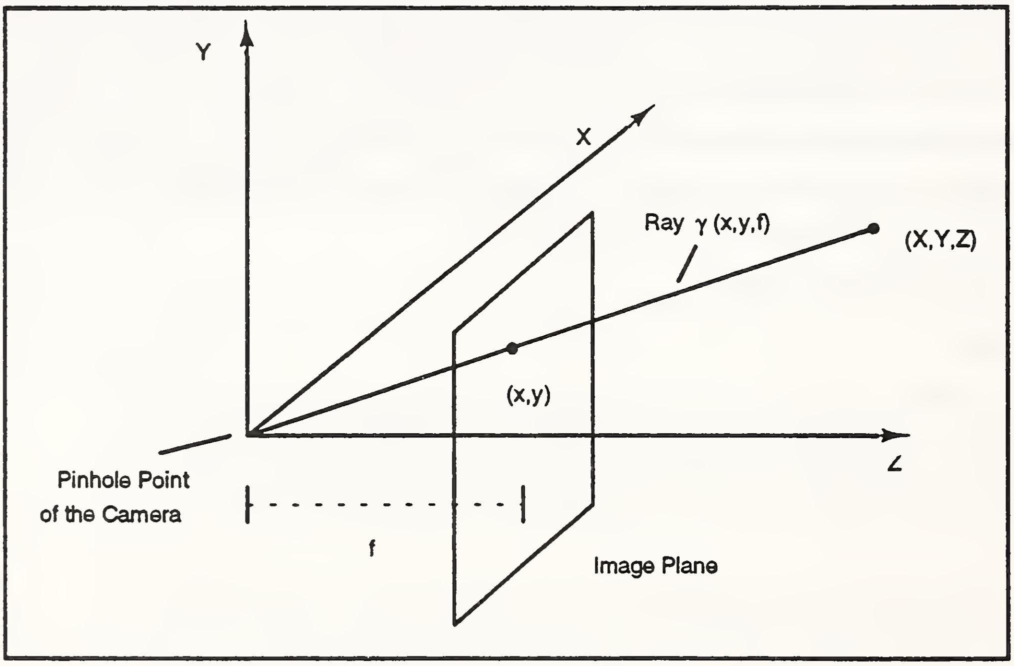


Fig. 3 The perspective projection

For any three dimensional point (X,Y,Z) its perspective projection (x,y) satisfies

$$x = f\frac{X}{Z}\ ,\ y = f\frac{Y}{Z}. \qquad \text{... (3)}$$

Let R be the distance between the pinhole point of the camera and the point determined by the intersection of the plane $AX + BY + Cf + D = 0$ with the ray λ(x,y,f), (Figure 3). This point is the three-dimensional point whose perspective projection is (x,y). Therefore,

$$R^2 = X^2 + Y^2 + Z^2. \qquad \text{... (4)}$$

It can be shown that [17] :

$$R^2 = \frac{x^2+y^2+f^2}{f^2}\frac{D^2 f^2}{(Ax+By+Cf)^2} = \frac{D^2}{(n^T\xi)^2} \qquad \text{... (5)}$$

where $n = \begin{pmatrix} A \\ B \\ C \end{pmatrix}$, (n^T is the transpose of n),

and $\xi = \frac{1}{\sqrt{x^2+y^2+f^2}}\begin{pmatrix} x \\ y \\ f \end{pmatrix}$

are both unit length vectors. **n** denotes the surface normal at (x,y,z).

Let the density of texture primitives on the plane $AX+BY+CZ+D = 0$ be k primitives per unit area. The density γ of texture primitives in the image in the area formed by the solid angle Ω is then given by [17] :

$$\gamma = \frac{kD^2}{f}\frac{1}{(n^T\xi)^3\sqrt{x^2+y^2+f^2}}. \qquad \text{... (6)}$$

This is the relationship between the texture primitive density γ at position (x,y) and the surface normal **n** of the textural surface.

3.3 Change of texture density and Looming

We need to establish a relationship between the temporal change of the texture primitive density γ and time rate of change of range of the surface.

Using Equation (5)

$$D^2 = R^2(n^T\xi)^2. \qquad \text{... (7)}$$

Combining Equations (6) and (7) yields :

$$\gamma = \frac{kR^2}{f} \frac{1}{(n^T\xi)\sqrt{x^2+y^2+f^2}} \qquad \text{... (8)}$$

or

$$R^2 = \frac{\gamma f n^T \xi \sqrt{x^2+y^2+f^2}}{k}. \qquad \text{... (9)}$$

If the camera continuously keeps the feature point on the textural surface fixated at the center of its screen, then x=y=0. This would be true for any motion of the camera as long as the point on the surface is always in fixation (Figure 4).

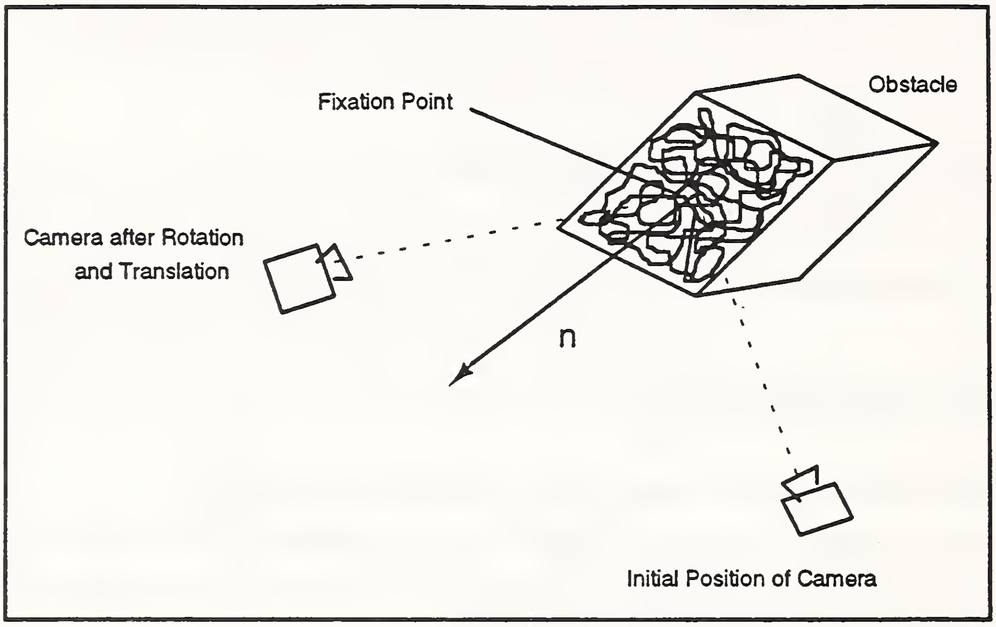

Fixation Point
Obstacle
Camera after Rotation and Translation
n
Initial Position of Camera

Fig. 4 Camera after rotation and translation

Therefore,

$$R^2 = \frac{\gamma f^2 n^T \xi}{k}. \qquad \text{... (10)}$$

where $\xi = \frac{1}{\sqrt{x^2+y^2+f^2}} \begin{pmatrix} x \\ y \\ f \end{pmatrix} = \frac{1}{f} \begin{pmatrix} 0 \\ 0 \\ f \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$

However,

$$n^T\xi = (ABC)\begin{pmatrix}0\\0\\1\end{pmatrix} = C = \cos\theta .$$

Hence,

$$R^2 = \frac{\gamma f^2 C}{k}. \qquad \text{... (11)}$$

Differentiating both sides of Equation (11) with respect to time yields:

$$2R\frac{dR}{dt} = \frac{f^2}{k}[C\frac{d\gamma}{dt} + \gamma\frac{dC}{dt}].$$

And dividing both sides by R^2 we get:

$$\frac{\frac{dR}{dt}}{R} = \frac{1}{2}(\frac{\frac{d\gamma}{dt}}{\gamma}) + \frac{1}{2}(\frac{\frac{dC}{dt}}{C}). \qquad \text{... (12)}$$

However, looming is defined as [15]:

$$L = -\frac{\frac{dR}{dt}}{R}.$$

Therefore, using Equations (1) and (12)

$$\boxed{L = -\frac{1}{2}(\frac{\frac{d\gamma}{dt}}{\gamma}) + \frac{1}{2}(\tan\theta)\frac{d\theta}{dt}.} \qquad \text{... (13)}$$

This expression can be expressed in a logarithmic form:

$$\boxed{L = \frac{d}{dt}\ln(\frac{\gamma_0}{\gamma}\cos\theta)} \qquad \text{... (14)}$$

where γ_0 is a constant.

In Equation (13), γ and $d\gamma/dt$ signify the image texture density and its time rate of change, respectively. θ is the angle between the surface normal and the optical axis of the camera. $d\theta/dt$ is the angular velocity of the camera. Therefore, since $\frac{d\gamma}{dt}/\gamma$ can be measured from the image, if θ can be calculated (hence $d\theta/dt$ can be obtained), looming value of the textured surface can be obtained using expression (13). The expression is valid for a general translational and rotational motion of a fixated camera and using computer simulation it has been verified.

3.4 Calculating θ from texture

We present here a simple idea that is computationally inexpensive, yet provides a good approximation of slant θ, that determines the relative orientation of the local planar surface with respect to the camera.

Assume that a very small patch on the textural surface has the surface normal **n** and its range with respect to a fixating camera is **R**. The angle between the two vectors **n** and **R** is θ (see Figure 5).

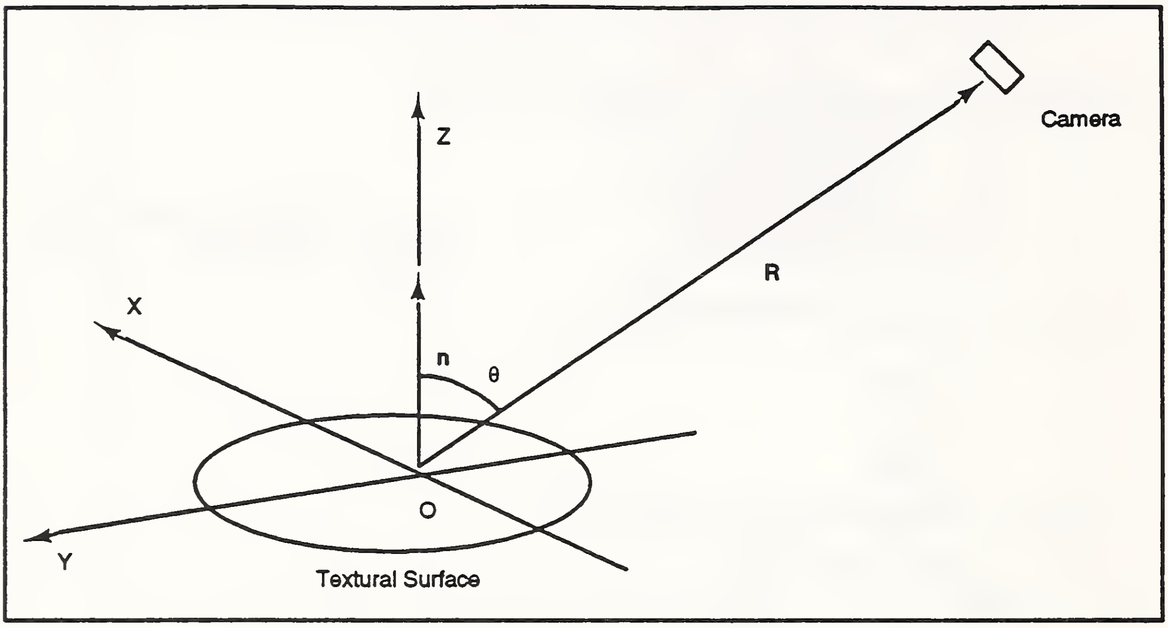


Fig. 5 Orientation of the surface

Without the loss of generality we can assume that **n**, **R** and the Y axis lie on the same plane and the X and Y axis lie on the patch. Let **R** be very large compared to the size of the patch.

The 1-D density of texture primitives on the surface, along a line, is given by $\frac{dT}{dl}$, where T is the number of texture features along a line and l is the length of the line.

If the surface has a uniform density of the texture primitives, then the minimum visible texture per unit angle due to a linear projection of a radial line is given by

$$D_{\min} \approx \frac{\frac{dT}{dl}\Delta l}{R} \qquad \text{... (15)}$$

and the maximum visible texture per unit angle due to a linear projection for a line perpendicular to the former is given by

$$D_{\max} \approx \frac{\frac{dT}{dl}\Delta l}{R\cos\theta} \qquad \text{... (16)}$$

(if $\Delta l \rightarrow 0$ then Equations (15) and (16) become equalities).

Hence if $D_{\min}$ and $D_{\max}$ are measured from the image, cosθ can be calculated by dividing Equation (15) by Equation (16).

$$\cos\theta \approx \frac{D_{\min}}{D_{\max}}. \qquad \text{... (17)}$$

A practical method of calculating $D_{\min}$ and $D_{\max}$ involves measuring the density of the texture primitives along several radial lines centered at the fixation point, comparing their values and finding the minimum and the maximum along two orthogonal lines.

4. Results

Figure (6) shows the setup for measuring looming. A camera attached to a Coordinate Measuring Machine (CMM) can be translated to a very high accuracy in the three-dimensional space by sliding it in the X, Y or Z directions. The camera is moved in steps towards the textural surface of a flat stationary object and the range between the camera and the surface is recorded. The initial range is 1000 mm and each step is 20 mm long. Using this range data the looming value is calculated from a discrete approximation of Equation (1). This is later used as the ground truth. Next a small window is created on the image around the camera fixation point and the texture density is measured. First the texture density is measured in the whole window which, as it changes with the changing range, provides one with dγ/dt. Next, the density in the radial directions are measured to calculate θ, the local slant of the surface, from Equations (15), (16) and (17). The object is slowly rotated to obtain dθ/dt. These values are used in Equation (13) to calculate the looming and the value is compared with the ground truth. To ease the measurement of texture density the image inside the window is passed through a Sobel operator. Weak and spurious texture lines are removed by a threshold, resulting in a binary image.

Texture primitives are counted by detecting the gray level changes in the window and also along the radial lines.

Figures (7) and (8) show the changes in the texture density in the image as the camera moves towards two different textures printed on white background and put on a flat object. Figures (9) through (14) illustrate the results by comparing the calculated looming values with the ground truth for different textures. In the first set of experiments in Figures (9) through (11) the camera optical axis and the surface normal were approximately parallel, hence $\theta \approx 0$. The camera underwent only translation towards the surface, therefore $d\theta/dt=0$. The textures used in the subsequent experiments were taken from Brodatz [24]. In the second set of experiments the surface was slanted at 0^0, 30^0 and 45^0. Figures (12) and (13) show this result for camera translation. Since there was no rotation of the camera $d\theta/dt=0$. Finally, Figure (14) illustrates the situation when the fixated camera had both translation and rotation and the surface was tilted. Instead of rotating the camera the surface was rotated at the rate of 1^0 in each step. To reduce noise every three data points have been averaged into one. Since our method of calculation of θ is approximate, $d\theta/dt$ calculated from this θ is usually noisy. It is assumed that $d\theta/dt$ (in this case 1^0/step) is available from another source (in practice it may be obtained from an encoder, a tachometer or a gyroscope attached to the fixated camera). The overall results are encouraging.

Two important issues need to be mentioned in this context. Any measurement in a texture is prone to an error because of subtextures. As a camera moves closer to a surface it captures more subtexture details. Also the accuracy in any texture analysis is dependent on the effects of the three-dimensional relief on the observed texture. In the approach that is presented here the analytical expressions had an assumption that the surfaces had a finite number of visible textures that did not have any three-dimensional relief.

Another source of error was the assumption that the texture density around the fixation point was uniform. The window size was considerable (200X200 pixels) while actually it needs to be very small, especially to use the Equation (15) and (16) to calculate $\tan\theta$. Also, the distance between the pinhole point and the image plane in the camera is known only approximately. And finally, focusing affects the texture density in an image. A camera with an automatic focusing will provide more accurate results.

5. Implementation of Obstacle Avoidance

A miniature camera is attached to a six-degrees-of-freedom flight simulator that allows autonomous vision-based navigation in a miniature environment. The simulator has been obtained by modifying a Gantry Robot. The modification is such that the robot's main controller is not being used, instead it is controlled directly by an 80486 based personal computer that supplies analog signals to the six control loops, each corresponding to a *velocity* input. This new control configuration of the robot allows continuous and smooth motion of the different axes.

Figure (15) shows the experimental setup. Artificial texture is painted in black on white cylindrical obstacles. The simulator starts from one end of the platform and needs to reach the other end through these obstacles. During its travel the end-effector rotates continuously causing the camera horizontally scan the scene in the front. Every time the camera detects an obstacle it fixates on it for a short period of time to calculate the texture density. The detection is done by calculating the texture in a small window as the camera scans the 3-D space. If the texture density in the window is higher than a threshold it is taken as part of an obstacle. Figure (16) shows the view of the camera as the flight simulator moves towards the obstacles. Looming of each obstacle is calculated from the texture of the surface, using the Equation (13), substituting $\theta \approx 0$ and $d\theta/DT \approx 0$. The control of the simulator is implemented in a closed loop following a simple algorithm described in [16] where the control objective is to minimize the threat emanating from each obstacle on the path. The looming values of the obstacles are sorted and one with the highest looming is considered the highest threat and the robot changes its course accordingly. In that process the clearance between two obstacles is also measured to maintain a safe distance from each of them. Figures (17 a.) through (17 c.) illustrate the results. The camera, after spotting the three cylindrical objects on its path fixates on them and calculates the individual looming values. When the looming of the one at the front increases and crosses a threshold, the simulator changes its course and travels between two objects, maintaining a safe clearance.

6. Conclusions

A new method to calculate the visual cue looming has been presented. The approach is based on the texture density and its change in the image due to a decreasing distance between a fixated camera and a surface. The surface has a finite number of visible textures. It is assumed that the small area around the fixation point is locally planar and the texture density on the surface is constant in that region. The method is simple, computationally inexpensive and practical and uses regular gray level images. It is texture independent (as long as the texture is visible and finite). The camera can undergo both translation and rotation and the surface can be tilted. The approach uses a single camera and utilizes *visual fixation.* The results illustrate how the measured values of looming stay close to the actual values. A camera with a good resolution and automatic focusing will improve the results significantly. Finally, looming has been used in the sense of a threat of collision, to navigate in an unknown environment. The results show that the approach can be used in real time obstacle avoidance without almost any a-priori knowledge.

Acknowledgment

The authors would like to thank Martin Herman, Ernie Kent and James Albus of NIST for useful discussions regarding the looming approach and P.S. Neelakanta from FAU for reviewing this paper and suggesting useful comments.

References

[1] Beek, P.J., "Perception-action coupling in the young infant : An appraisal of von Hofsten's research programme," In M.G. Wade and H.T.A. Whiting (Eds.), *Motor Development in Children: Aspects of coordination and control,* Dordrecht, The Netherlands: Martinus-Nijhoff, pp. 187-196, 1986.

[2] Bower, T.G.R, Broughton, J.M. and Moore, M.K., "The coordination of visual and tactical input in infants," *Perception and Psychophysics,* 8, pp. 51-53, 1970.

[3] Schiff, W, Carviness, J.A. and Gibson, J.J., "Persistent fear responses in rhesus monkeys to the optical stimulus of looming," *Science* 136, pp. 982-983, 1962.

[4] Lee, D.N., Young, D.S., Reddish, D.E., Lough, S. and Clayton, T.M.H., "Visual timing in hitting an accelerating ball," *Quarterly Journal of Experimental Psychology,* 35A, pp. 333-346, 1983.

[5] Savelsbergh, G.J.P., Whiting, H.T.A. and Bootsma, R.J., "Grasping Tau," *Journal of Experimental Psychology: Human Perception and Performance,* 7, pp. 795-810, 1991.

[6] Blostein, D. and Ahuja, N., "Shape from texture: Integrating texture-element extraction and surface estimation," *IEEE Transactions on PAMI*, Vol. 11, No. 12, pp. 1233-1251, December 1989.

[7] Lee, D.N., "A theory of visual control of braking based on information about time-to-collision," *Perception,* 5, pp. 437-459, 1976.

[8] Nelson, R.C. and Aloimonos, J., "Obstacle avoidance using flow field divergence," *IEEE Trans. on PAMI,* Vol. 11, No. 10, October 1989.

[9] Nelson, R.C. and Aloimonos, J., "Finding motion parameters from spherical flow fields," Computer Science Technical Report Series, University of Maryland, College Park, Maryland, 20742.

[10] Cipolla, R. and Blake, A., "Surface orientation and time to contact from image divergence and deformation," *Proc. of the European Conf. on Computer Vision*, pp. 187-202, 1992.

[11] Tistarelli, M. and Sandini, G., "On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow," *IEEE Transactions on PAMI*, Vol. 15, No. 4, pp. 401-410, April 1993.

[12] Murphy, R.R., "Strategy for the fine positioning of a mobile robot using texture," *SPIE,* Vol. 1195, Mobile Robots VI, pp. 267-279, 1991.

[13] Aloimonos, J., Weiss, I. and Bandopadhaya A., "Active Vision," *International Journal of Computer Vision,* 1(4), pp. 333-356, 1988.

[14] Aloimonos, J., "Purposive and qualitative active vision," *Proc. International Workshop on Active Control in Visual Perception,* Antibes, France, April, 1990.

[15] Raviv, D., "Visual looming," *SPIE,* Vol. 1825, Intelligent Robots and Computer Vision XI: Algorithms, Techniques and Active Vision, November, 1992.

[16] Joarder, K. and Raviv, D., "Autonomous obstacle avoidance using visual fixation and looming," *SPIE,* Vol. 1825, Intelligent Robots and Computer Vision XI: Algorithms, Techniques and Active Vision, November, 1992.

[17] Haralick, R.M. and Shapiro, L.G., "Computer and Robot Vision," Vol. 1, Addison-Wesley Publishing Company, pp. 483-493, 1992.

[18] Aloimonos, J. and Swain, M.J., "Shape from texture," *Proc. of the Ninth International Joint Conference on Artificial Intelligence,* pp. 926-931, Los Angeles, 1988.

[19] Aloimonos., J., "Detection of surface orientation from texture I: The case of planes," *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, pp. 584-593, 1986.

[20] Bajcsy, R. and Lieberman, L., "Texture gradient as a depth cue," *Computer Graphics Image Processing*, Vol. 5, pp. 52-67, 1987.

[21] Kanatani, K., "Detection of surface orientation and motion from texture by a stereological technique," *Artificial Intelligence*, Vol 23, pp. 213-237, 1984.

[22] Gibson, J., "Perception of the visual world," Boston, MA, Houghton Miffin, 1950.

[23] Krumm, J. and Shafer, S., "Shape from periodic texture using the spectrogram," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 284-289, June, 1992.

[24] Brodatz, P., "Textures - a photographic album for artists and designers," Dover Publications, Inc.,

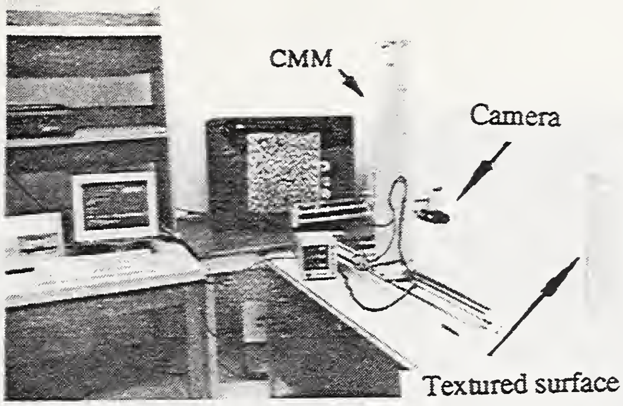


Fig. 6 Experimental setup for measuring looming

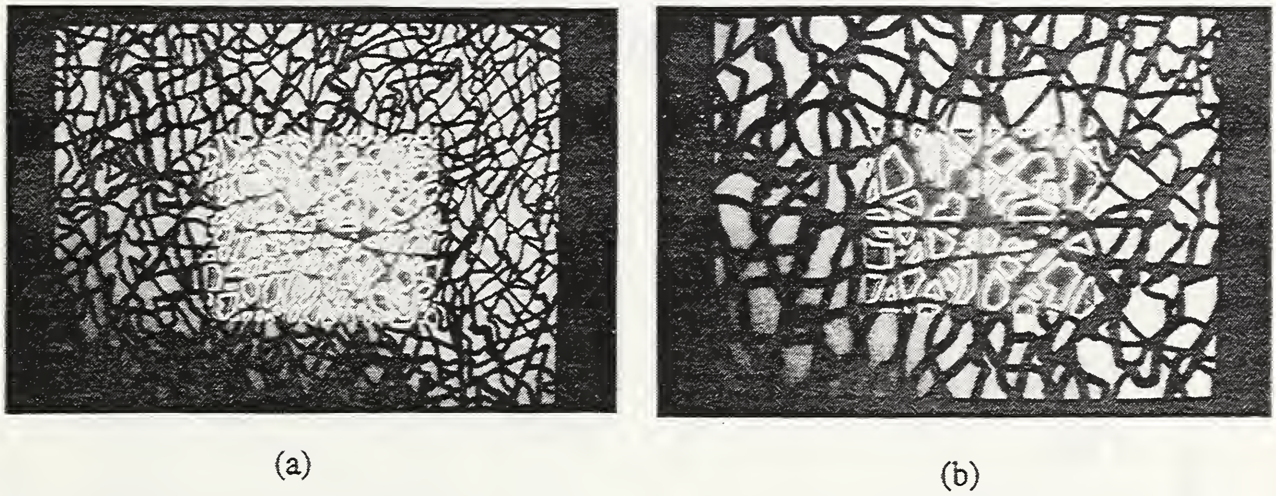


Fig. 7 Texture I : Change in texture density

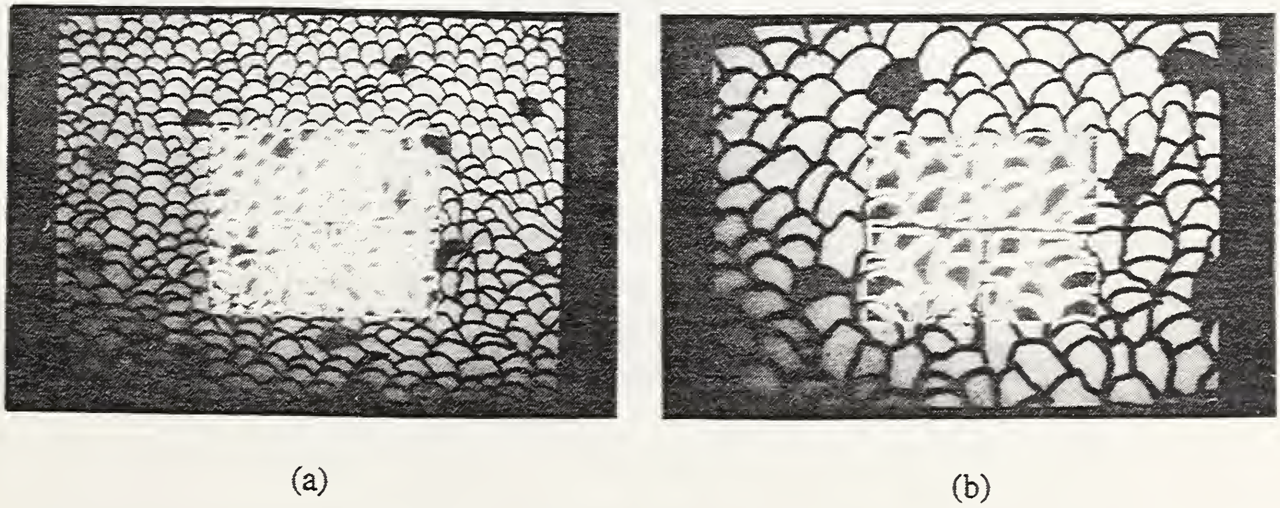


Fig. 8 Texture II : Change in texture density

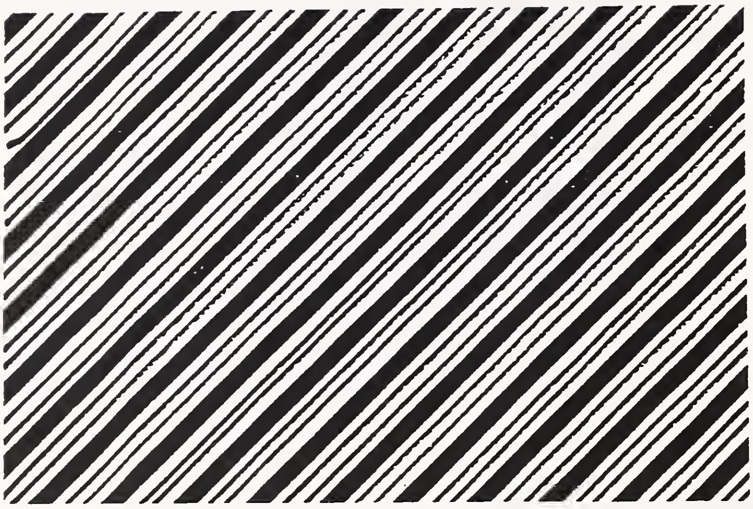


Fig. 9 (a) Texture III

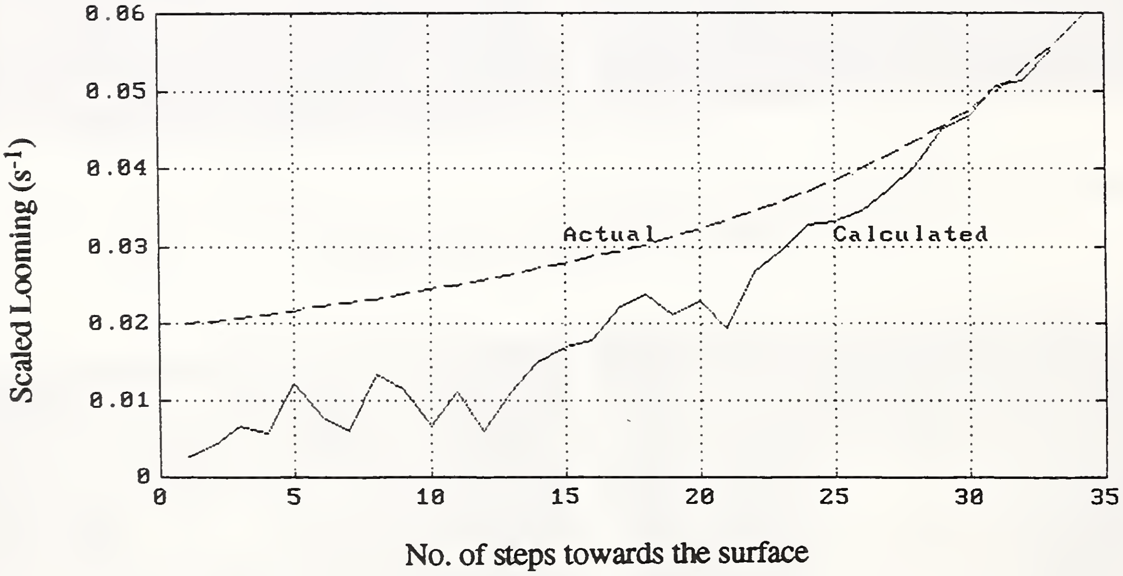


Fig. 9 (b) Calculated and actual looming values (Slant 0^0)

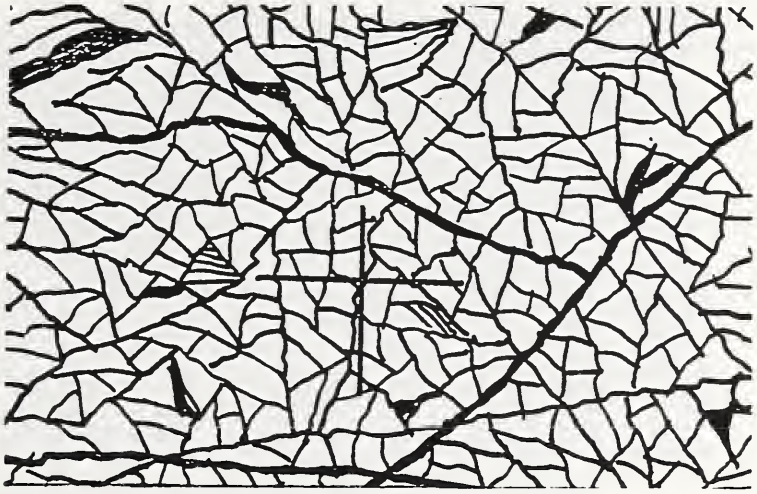


Fig. 10 (a) Texture IV

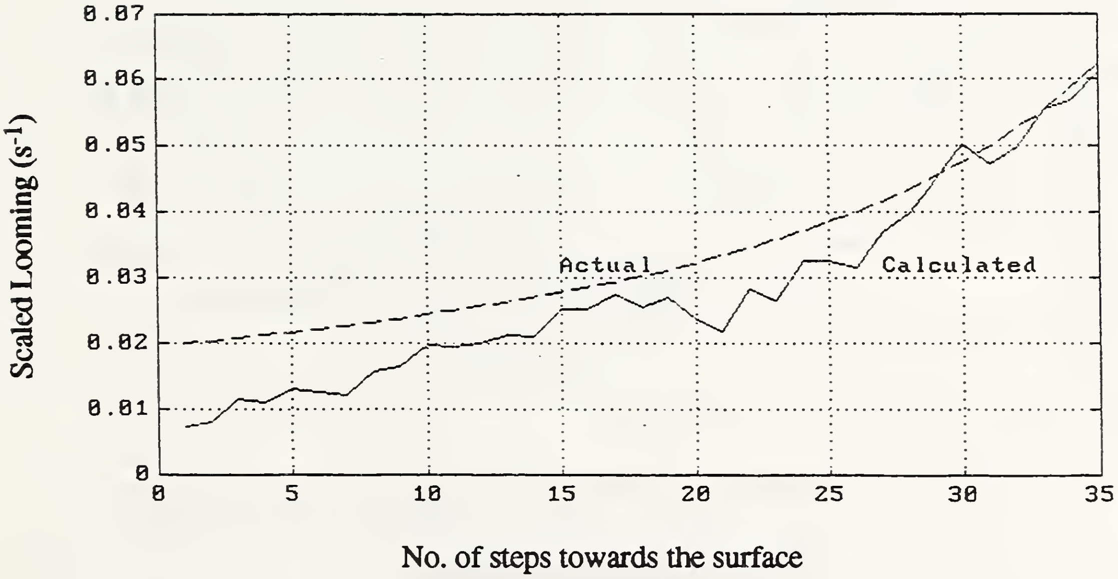


Fig. 10 (b) Calculated and actual looming values (Slant 0^0)

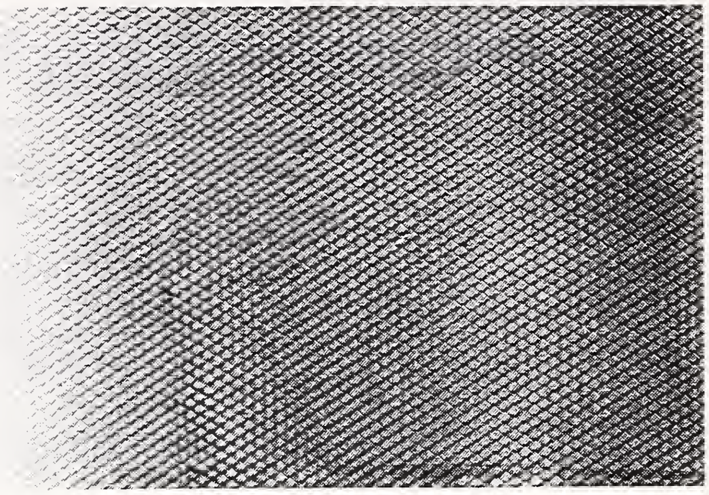
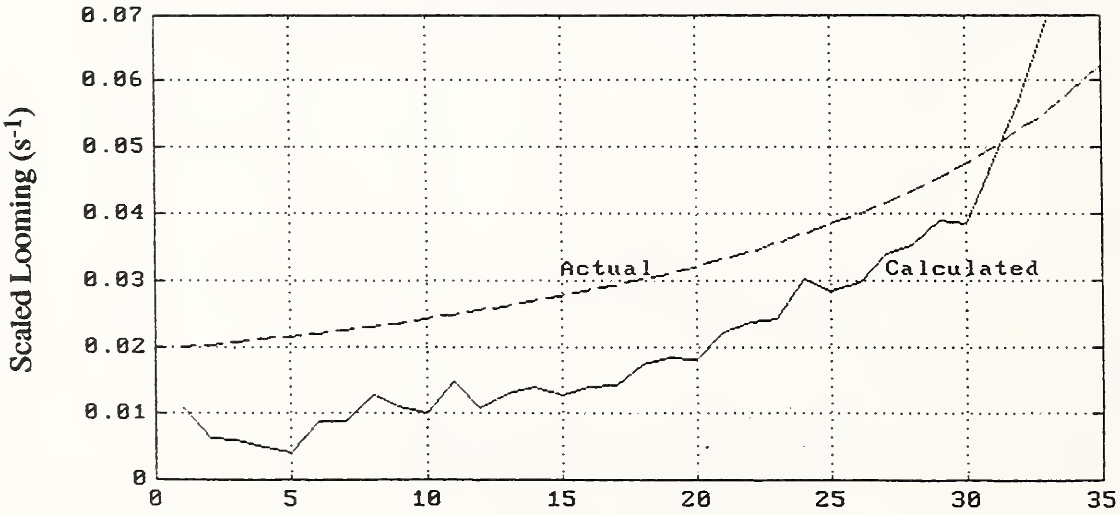


Fig. 11 (a) Texture V (Steel net)



No. of steps towards the surface

Fig. 11 (b) Calculated and actual looming values (Slant 0^0)

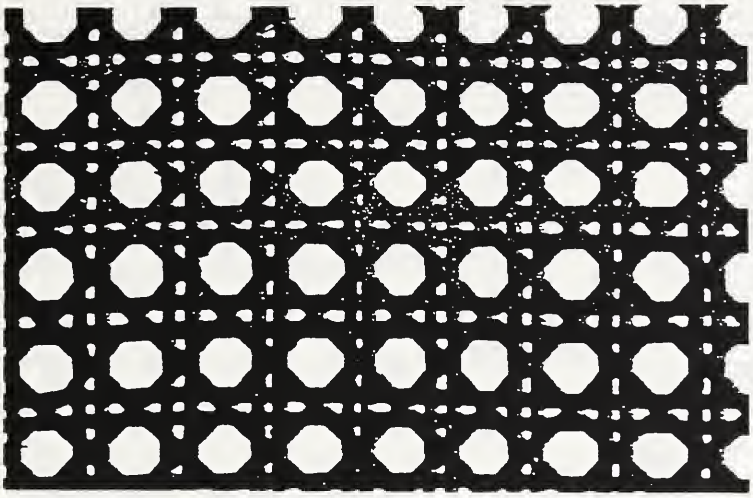


Fig. 12 (a) Texture VI (Cane)

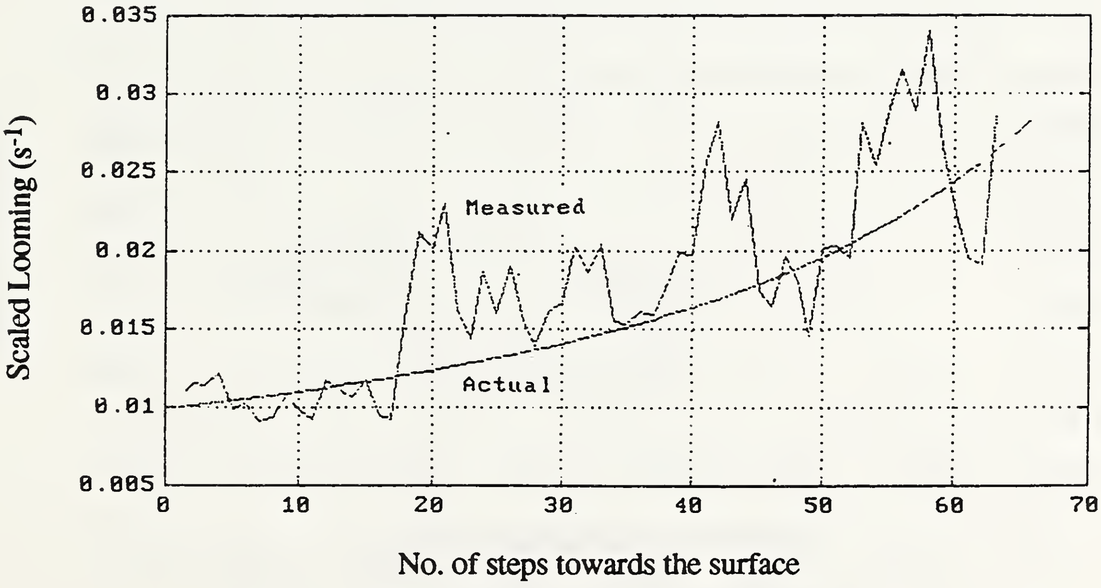


Fig. 12 (b) Calculated and actual looming values (Slant 0^0)

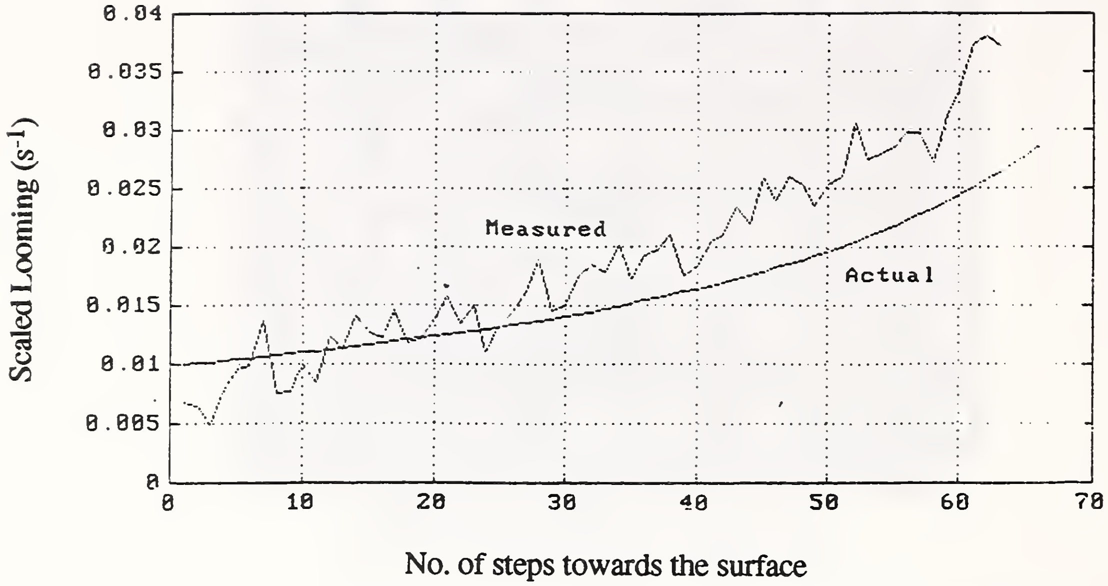


Fig. 12 (c) Calculated and actual looming values (Slant 30^0)

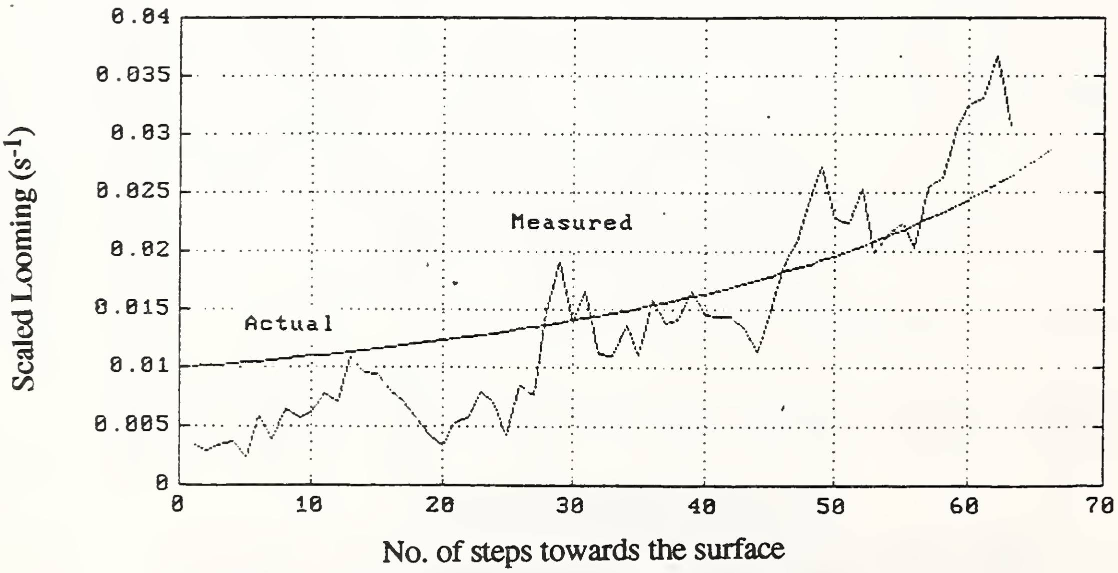


Fig. 12 (d) Calculated and actual looming values (Slant 45^0)

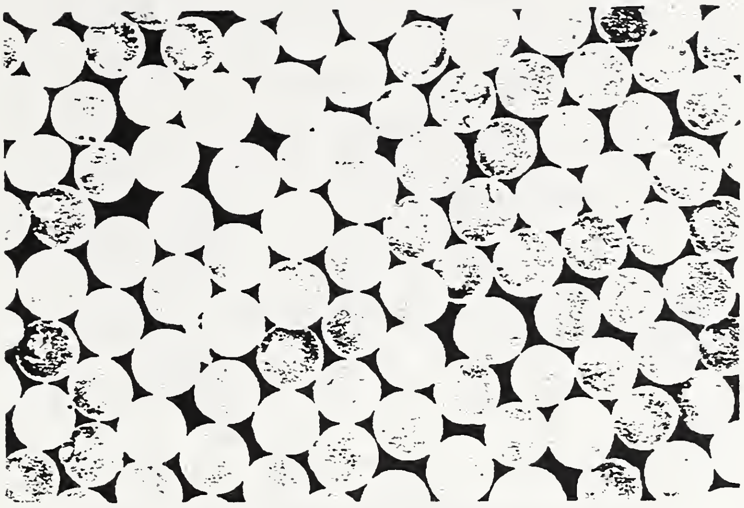


Fig. 13 (a) Texture VII (Plastic pellets)

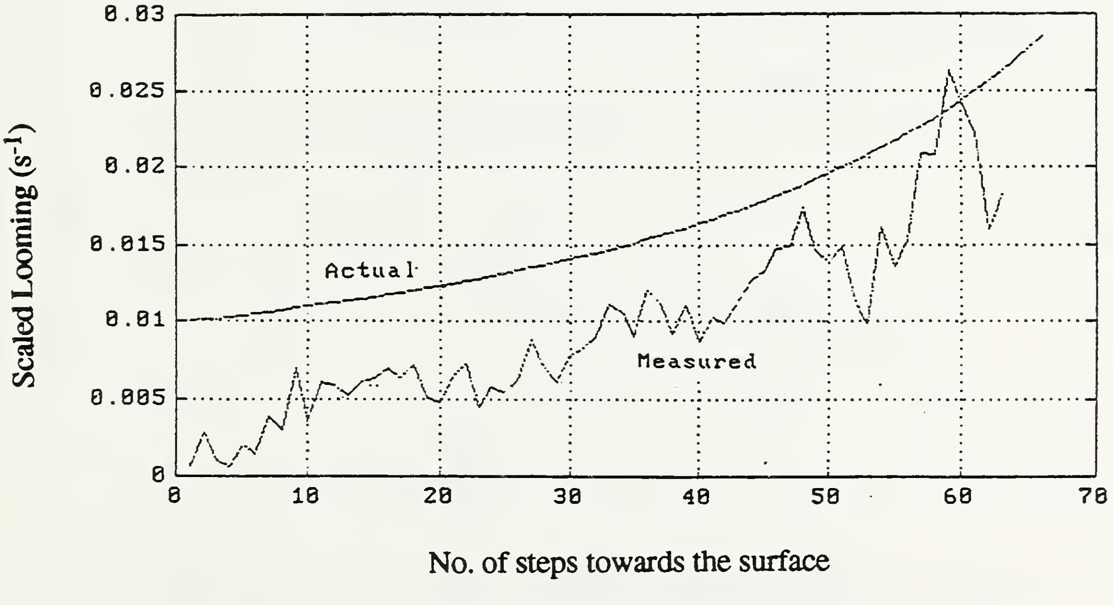


Fig. 13 (b) Calculated and actual looming values (Slant 0^0)

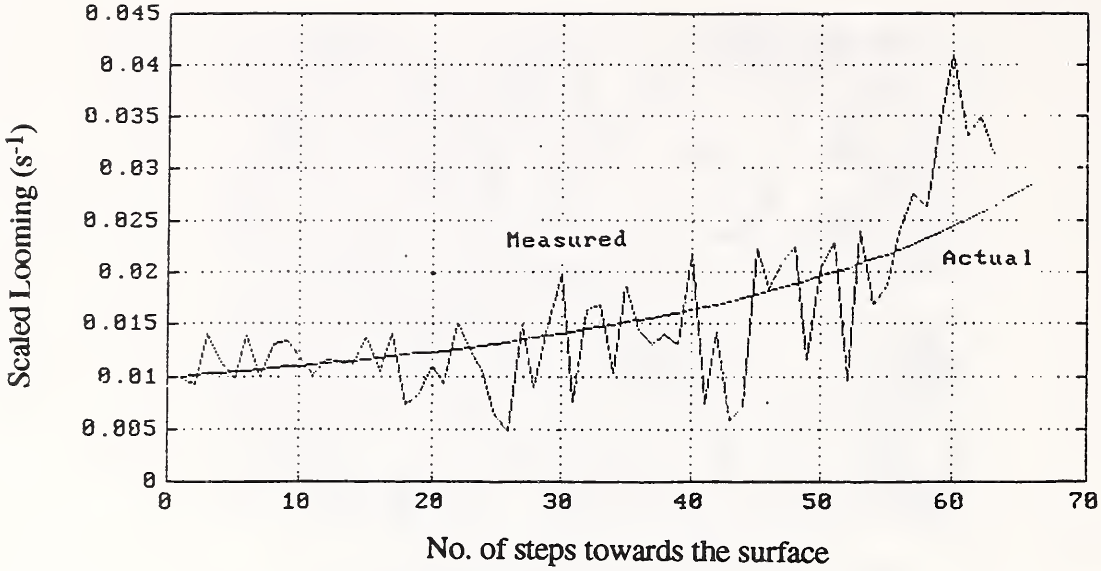


Fig. 13 (c) Calculated and actual looming values (Slant 30^0)

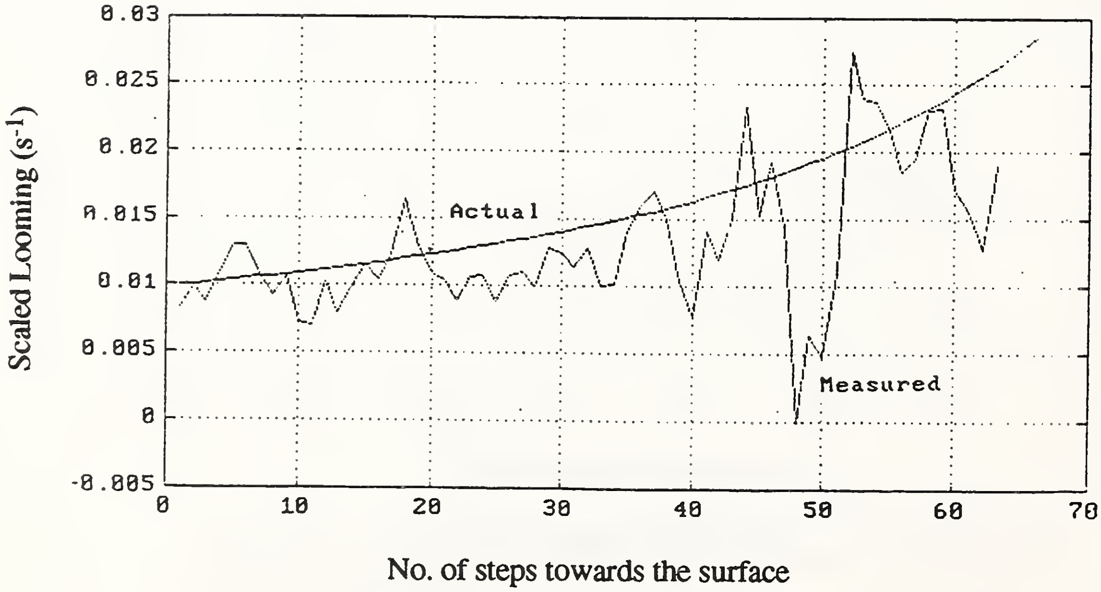


Fig. 13 (d) Calculated and actual looming values (Slant 45^0)

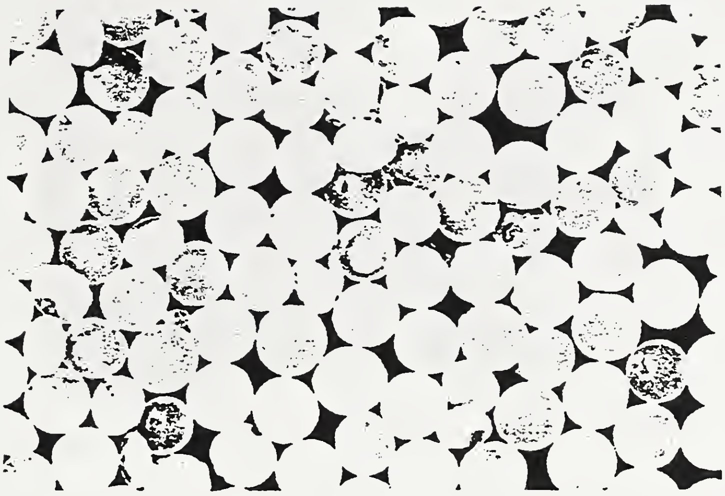


Fig. 14 (a) Texture VII (Plastic pellets)

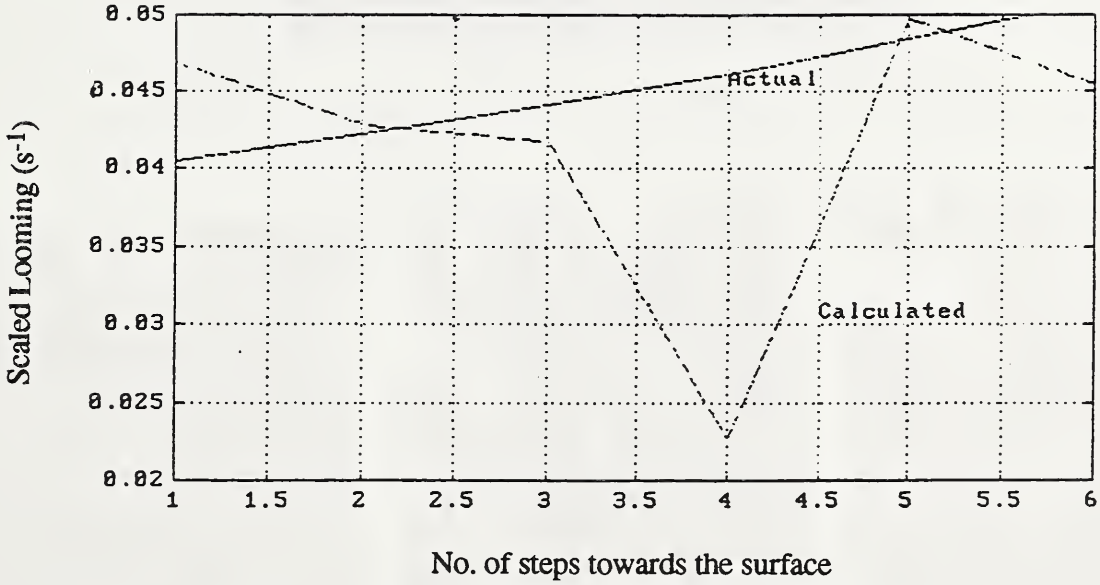


Fig. 14 (b) Looming values for tilted, rotating surface

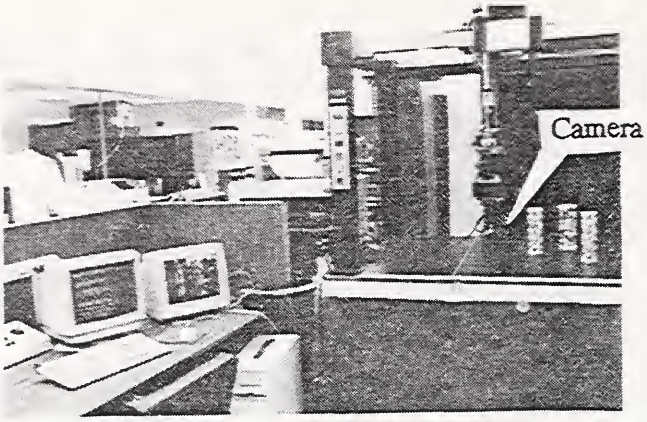


Fig. 15 Flight simulator for obstacle avoidance

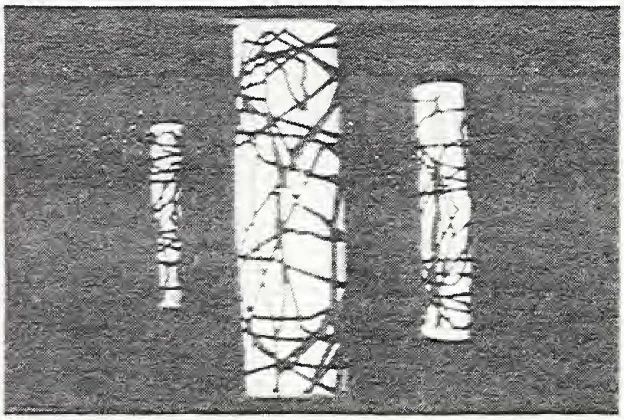


Fig. 16 View of the miniature camera

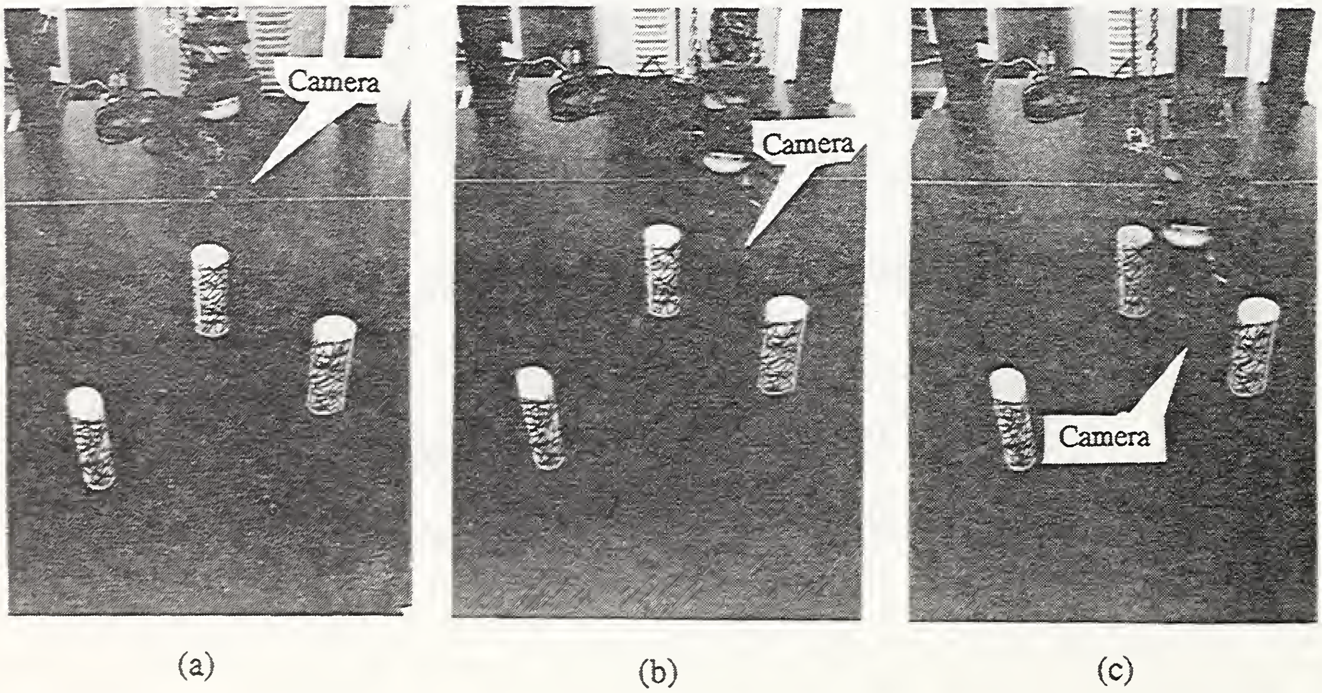


Fig. 17 The camera changing its course during obstacle avoidance